

Lesson 3

Understanding General Software Development

Knowledge Assessment

Multiple Choice

Circle the letter that corresponds to the best answer.

1. The product that you are developing is not yet finished, but you would like to release the product to a wider customer audience for feedback and testing. Under which of the following testing levels would this activity fall?
 - a. Integration testing
 - b. System testing
 - c. Acceptance testing**
 - d. Regression testing
2. The testers of a software application have access to its source code and they plan to write test cases that ensure that the methods return correct values. Which of the following testing levels will this activity fall under?
 - a. Integration testing
 - b. Unit testing**
 - c. Alpha testing
 - d. Beta testing
3. Which of the following data structures allows direct access to all of its items?
 - a. Array**
 - b. Stack
 - c. Queue
 - d. Linked list
4. Which of the following activities in the application lifecycle is used by an architect to create the technical blueprint of a system?
 - a. Requirements analysis
 - b. Design**
 - c. Development
 - d. Maintenance
5. In your application, you are using a queue data structure to manipulate information. You need to find which data item will be processed next, but you don't want to actually process that data item yet. Which of the following queue operations will you use?
 - a. Enqueue
 - b. Dequeue
 - c. Peek**

- d. Contains
6. You are developing a program that requires you to track the method calls. You can only invoke one method at a time. However, a method call may in turn invoke other methods. When a method ends, it returns control back to the calling method. Which data structure should you use to keep track of these method calls?
- a. Queue
 - b. Array
 - c. Linked list
 - d. Stack**
7. You are developing a program that simulates a job queue. Often, the jobs come faster than you can process them, and in such case, the jobs wait for their turn to be processed. You need to make sure that the job that arrived first is the first to be processed as well. Which of the following data structures is best suited for this requirement?
- a. Array
 - b. Queue**
 - c. Linked list
 - d. Stack
8. You write the following code in a program:
- ```
int[] numbers = {2, 3, 1, 4};
numbers[2] = 4;
```
- What will be the contents of the array after the second statement is executed?
- a. {2, 4, 1, 4}
  - b. {2, 3, 4, 4}**
  - c. {2, 4, 1, 2}
  - d. {4, 3, 1, 4}
9. You are developing a program that performs frequent insert and delete operations on the data. Your requirement also dictates the capability to access previous and next records when the user presses the previous or next button. Which of the following data structures will best suit your requirements?
- a. Array
  - b. Circular linked list
  - c. Linked list
  - d. Doubly linked list**
10. You are developing a program that performs frequent insert and delete operations on the data. The data items need to be accessed like a stack with last-in, first-out functionality. Your

solution must require as little memory as possible. Which of the following data structures will best suit these requirements?

- a. Array
- b. Circular linked list
- c. Linked list**
- d. Doubly linked list

### Fill in the Blank

---

**Complete the following sentences by writing the correct word or words in the blanks provided.**

1. In **white-box** testing, testers use their knowledge of system internals to assess the system.
2. Usually, with every new fix, software testers run a battery of **regression tests** to make sure that all functionality that was known to be working is still working.
3. The BubbleSort algorithm uses a series of **comparison** and **swap** operations to arrange the elements of a list in the correct order.
4. A(n) **stack** is a collection of items in which the last item added to the collection is the first one to be removed.
5. **Requirements analysis** is the process of determining the detailed business requirements for a new software system.
6. A linked list is a collection of nodes such that each node contains a(n) **reference (or link)** to the next node in the sequence.
7. The **enqueue** operation adds an item to a queue, whereas the **dequeue** operation removes an item from a queue.
8. The QuickSort algorithm uses **partitioning** and comparison operations to arrange the elements of a list in the correct order.
9. A(n) **business analyst** is responsible for analyzing business needs and converting them into requirements that can be executed by the development team.
10. Alpha testing and beta testing both are part of the **acceptance** testing of a system.

### Competency Assessment

#### Project 3-1: Using Arrays

---

You are writing a program that uses a two-dimensional array. The array has four rows and five columns. You need to print the largest element in each row of the array. How would you write such a program?

- 1. Create a new project based on the Console Application template. Name the project Project03\_01.**
- 2. Replace the code for the Program.cs file with the following:**

```
using System;
namespace Project03_01
{
 class Program
 {
 static void Main(string[] args)
 {
 int[,] numbers = new int[,]
 {
 { 11, 7, 50, 45, 27 },
 { 18, 35, 47, 24, 12 },
 { 89, 67, 84, 34, 24 },
 { 67, 32, 79, 65, 10 }
 };

 for (int row = 0;
 row < numbers.GetLength(0);
 row++)
 {
 Console.WriteLine(
 "Maximum number in the row {0}: {1}",
 row, FindMax(row, numbers));
 }
 Console.WriteLine(
 "Press any key to continue...");
 Console.ReadKey();
 }

 static int FindMax(int row, int[,] numbers)
 {
 int max = numbers[row, 0];
 for (int column = 0;
 column < numbers.GetLength(1);
 column++)
 {
 if (numbers[row, column] > max)
```

```
 max = numbers[row, column];
 }
 return max;
}
}
```

3. Select Debug > Start Debugging (or press F5) to run the project. You will see the following output displayed in the console window:

```
Maximum number in the row 0: 50
Maximum number in the row 1: 47
Maximum number in the row 2: 89
Maximum number in the row 3: 79
```

### Project 3-2: Using Queues

---

You are writing a program that uses two queues. The data in each queue is already in ascending order. You need to process the contents of both queues in such a way that the output is printed on the screen in sorted order. How would you write such a program?

1. Create a new project based on the Console Application template. Name the project Project03\_02.
2. Replace the code for the Program.cs file with the following:

```
using System;
using System.Collections;

namespace Project03_02
{
 class Program
 {
 static void Main(string[] args)
 {
 Queue first = new Queue();
 first.Enqueue(7);
 first.Enqueue(11);
 first.Enqueue(45);
 first.Enqueue(50);
```

```
Queue second = new Queue();
second.Enqueue(12);
second.Enqueue(32);
second.Enqueue(65);
second.Enqueue(67);

ProcessInOrder(first, second);

Console.WriteLine(
 "Press any key to continue...");
Console.ReadKey();
}

static void ProcessInOrder(Queue first,
 Queue second)
{
 while (first.Count > 0 || second.Count > 0)
 {
 if (first.Count == 0)
 {
 Console.WriteLine(second.Dequeue());
 continue;
 }

 if (second.Count == 0)
 {
 Console.WriteLine(first.Dequeue());
 continue;
 }

 if ((int) first.Peek()
 >= (int) second.Peek())
 {
 Console.WriteLine(
 second.Dequeue());
 }
 }
}
```

```
 else
 {
 Console.WriteLine(first.Dequeue());
 }
 }
}
}
```

3. Select Debug > Start Debugging (or press F5) to run the project. Notice that the numbers are displayed in ascending order in the console window.

## Proficiency Assessment

### Project 3-3: Using Stacks

You are writing a program that uses two stacks. The data in each stack is already in descending order. You need to process the contents of both stacks in such a way that the output is printed on the screen in ascending order. How would you write such a program?

1. Create a new project based on the Console Application template. Name the project Project03\_03.
2. Replace the code for the Program.cs file with the following:

```
using System;
using System.Collections;

namespace Project03_03
{
 class Program
 {
 static void Main(string[] args)
 {
 Stack first = new Stack();
 first.Push(50);
 first.Push(45);
 first.Push(11);
 first.Push(7);
```

```
Stack second = new Stack();
second.Push(67);
second.Push(65);
second.Push(32);
second.Push(12);

ProcessInOrder(first, second);

Console.WriteLine(
 "Press any key to continue...");
Console.ReadKey();
}

static void ProcessInOrder(Stack first,
 Stack second)
{
 while (first.Count > 0 || second.Count > 0)
 {
 if (first.Count == 0)
 {
 Console.WriteLine(second.Pop());
 continue;
 }

 if (second.Count == 0)
 {
 Console.WriteLine(first.Pop());
 continue;
 }

 if ((int)first.Peek()
 >= (int)second.Peek())
 {
 Console.WriteLine(
 second.Pop());
 }
 }
}
```



```
 else
 {
 Console.WriteLine(first.Pop());
 }
 }
}
}
```

3. Select Debug > Start Debugging (or press F5) to run the project. Notice that the numbers are displayed in ascending order in the console window.

### Project 3-4: Using Linked Lists

---

You are writing a program that stores a list of product names in a linked list. The user will enter a product name and your program needs to check whether the linked list contains the given product. How would you write such a program?

1. Create a new project based on the Console Application template. Name the project Project03\_04.
  2. Replace the code for the Program.cs file with the following:
- ```
using System;
using System.Collections.Generic;
```

```
namespace Project03_04
{
    class Program
    {
        static void Main(string[] args)
        {
            string[] words = {
                "Konbu", "Tofu",
                "Pavlova", "Chocolate",
                "Ikura" };
            LinkedList<string> list =
                new LinkedList<string>(words);

            Console.WriteLine("Enter product to find:");
            Console.WriteLine(
```

```
        "Type <end> to end the program");  
        string searchText = Console.ReadLine();  
        while (searchText != "<end>")  
        {  
            if (list.Contains(searchText))  
            {  
                Console.WriteLine(  
                    "The search text was found");  
            }  
            else  
            {  
                Console.WriteLine(  
                    "The search text was NOT found");  
            }  
        }  
  
        Console.WriteLine(  
            "Enter product to find:");  
        Console.WriteLine(  
            "Type <end> to end the program\n");  
        searchText = Console.ReadLine();  
    }  
}  
}
```

3. Select Debug > Start Debugging (or press F5) to run the project. Type a product name to find in the console window and press the Enter key. The program will display a message saying that the product was found or not found. Notice that the search is case sensitive. Type <end> and press Enter to end the program.